Terminal to Battery Processor Communication

Battery Processor to Terminal Communication

Command / Response Communication Protocol

```
        ( Power up. )
             │
             ▼
      ┌──────────────┐
      │ Initialize   │
      │ storage      │
      └──────────────┘
             │
   ┌─────────▶
   │         ▼
   │       ╱   ╲
   │      ╱Charge╲         Yes
   │      ╲applied?╲───────────────┐
   │       ╲     ╱                 │
   │         ╲ ╱                   ▼
   │          │ No              ╱‾‾‾‾╲    CHARGE
   │          ▼                │  A   │   FUNCTION
   │   ┌──────────────┐        ╲____╱
   │   │ Setup timer  │
   │   │ enter WAIT   │
   │   │ mode         │
   │   └──────────────┘
   │          │
   │          ▼
   │        ┌────┐⟲
   │        │WAIT│
   │        └────┘
   │          │        "WAKEUP"
   │          ▼
   │        ╱   ╲
   │ TIMER ╱WAKEUP╲
   └──────╱ SOURCE? ╲
           ╲     ╱
            ╲   ╱
              │ INTERRUPT
              ▼
      ┌──────────────┐
      │ Call receive │ (read command)
      │ subroutine.  │
      └──────────────┘
              │
              │ COMMAND TYPE
   ┌──────┬───┴───┬──────┬──────┐
  ∅x     1x      2x     3x     4x
   ▼      ▼       ▼      ▼      ▼
 ╱‾‾╲   ╱‾‾╲    ╱‾‾╲   ╱‾‾╲   ╱‾‾╲
│ B  │ │ C  │  │ D  │ │ E  │ │ F  │
 ╲__╱   ╲__╱    ╲__╱   ╲__╱   ╲__╱

 Reset/  Read    Set    Read   Deep cycle
 status  analog  alert  history
         value   points
```

BEST AVAILABLE COPY

B   Reset / status
~~TYPE~~ φx

φφ   φ1   φ2

Reset communications (error recovery)

Call transmit subroutine (STATUS)

Initialize history, default values

Setup Timer, enter WAIT mode

Wait

"Wakey"

Wakeup source? → INTERRUPT

TIMER

I   Communication function

Charge applied → Yes

No

A   Charge function

Read analog discharge current value

Compute ($Q_i =$) charge delivered

Update fuel guage

BEST AVAILABLE COPY

Charge Function

```
A

Read analog:
charge
voltage
  │
  ▼
Charge voltage within range?
  │No              │Yes
  ▼                ▼
High/low        Read analog:
limit?          temperature
  │High  │Low        │
  ▼      ▼           ▼
  H    Terminate   Table lookup:   (256 × 8)
       charge      charge current
       function    level
Error    │           │
condition ▼          ▼
         G        Output charge
                  level to port
```

High (from High/low limit → H)

Interrupt

```
Source?
  │Comm          │Timer
  ▼              ▼
  I          Increment
             software
             count
               │
               ▼
           Terminal count?
             │Yes      │No
             ▼         ▼
          Multiply   Return from
          time × current   interrupt
          = charge
             │
             ▼
         Table lookup:    (16 × 4)
         charge efficiency
         vs. temp
             │
             ▼
         Update fuel
         gauge
             │
             ▼
         Return from
         interrupt
```

Note: Processor does not enter WAIT condition during charge.

```
        C        Read analog
                 Type 1X

         ┌──────────┬──────────────┬──────────────┐
       10│        11│            12│            13│
   ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐
   │Read analog│ │Read analog│ │Read analog│ │Read analog│
   │channel 0: │ │channel 1: │ │channel 2: │ │channel 3: │
   │charge volt│ │discharge I│ │battery vol│ │temperature│
   └──────────┘ └──────────┘ └──────────┘ └──────────┘

              ┌──────────────┐
              │Call transmit │
              │  subroutine  │
              └──────────────┘

                    G


        D        Set Alert Points
                 Type 3X

              ┌──────────────┐
              │Call receive  │
              │  subroutine  │
              └──────────────┘

         ┌──────────┬──────────────┬──────────────┐
       20│        21│            22│
   ┌──────────┐ ┌──────────┐ ┌──────────┐
   │Set low   │ │Set low   │ │Set minimum│
   │reserve   │ │voltage   │ │capacity  │
   │alert point│ │alert point│ │alert point│
   └──────────┘ └──────────┘ └──────────┘

                    G
```
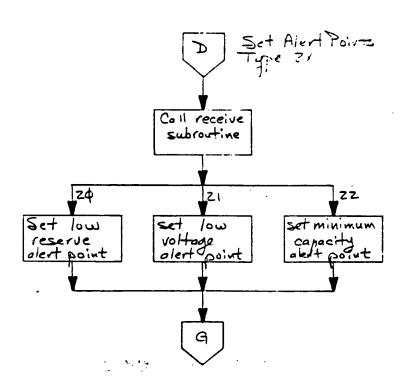
E Read history
   Type 3x

| 30 | 31 | 32 | 33 | 34 | 35 |

Send fuel
guage

Send max.
Jail. capacity

Send charge
cycle count
byte 1

Send deep
cycle count
byte 1

Send hours
used
byte 1

Send use
history
byte 1

Send charge
cycle count
byte 2

Send deep
cycle count
byte 2

Send hours
used
byte 2

Send use
history
byte 2

Send use
history
byte 3

Send use
history
byte 40

G

F | Deep Cycle function Type 40

Charge voltage within range? — No → H Error condition

Yes

Turn on discharge load

Read analog: battery voltage

next charge: >75Ω ?

battery voltage = 4.0V — No

turn off discharge load

Yes

Set new max. avail capacity

Deep cycle: consider: full charge cycle before deep discharge

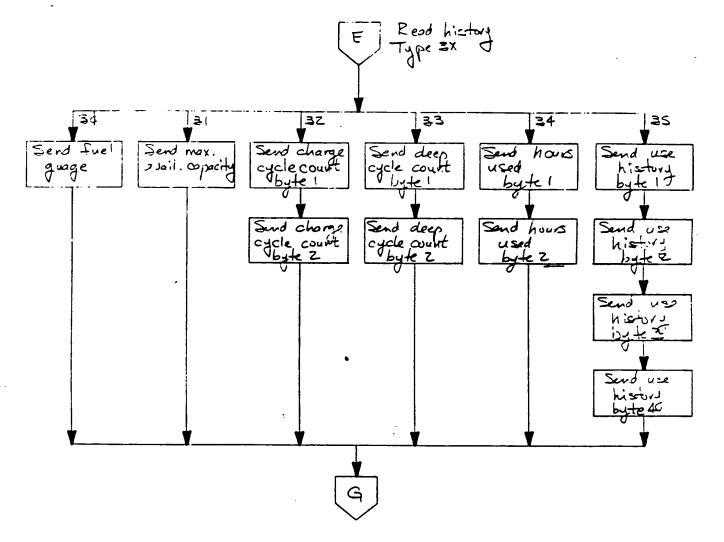Increment deep cycle count

Update DC: max. vs. present capacity

A | Charge function

Note: Processor does not enter WAIT condition during deep discharge

Interrupt

Source? — Comm → I

Timer

Increment software count

terminal count? — No → Return from Interrupt

Yes

Read analog: discharge current

Multiply time × current = charge (updating new C Vol~)

Return from Interrupt

I — Enter Communications
(Get Command)

BPCLK = φ — Yes

No

Sample
BPWDATA

BPWDATA = φ — No → Not a valid
STACT bit

Yes

Set bit
count
BC = φ

BPCLK = 1? — Yes

No

BPCLK = φ? — Yes

No

Read BPWDATA,
Set bit in
Rx char. buffer

Inc BC

BC = 1φ? — Yes
No

Parity O.K.? — No

Yes

Stop O.K.? — No

Send
ACK → Yes

Return

Data is in
receive
buffer

Flag error
condition
in accumulator

Return

-B8-

```
                                    ( Transmit subroutine
        ( Enter )                     data is in Tx Buffer )

           │
           ▼
      ┌─────────┐
      │ Set BC = 8 │                                ┌───────────────┐
      └─────────┘                                   │               │
           │                                        ▼               │
           ▼                                    ◇─────────◇  No      │
      ┌─────────┐                              ◇ BPCLK = φ ◇────────┤
      │ Compute │                               ◇─────────◇         │
      │ parity  │                                   │ Yes           │
      └─────────┘                                   ▼               │
           │  ◄──────────┐                          ─               │
           ▼             │                      ◇─────────◇  No      │
      ◇─────────◇        │                     ◇ BPCLK = 1 ◇────────┤
     ◇ BPCLK = 1? ◇ Yes  │                      ◇─────────◇         │
      ◇─────────◇────────┘                          │ Yes           │
           │ No                                     ▼               │
           ▼                                    ┌─────────┐         │
      ┌─────────┐                               │ Output  │         │
      │ Output  │                               │ parity  │         │
      │ start bit│                              └─────────┘         │
      └─────────┘                                   │  ◄────────────┤
           │   ◄──────────┐                         ▼               │
           ▼              │                     ◇─────────◇  No      │
      ┌─────────┐         │                    ◇ BPCLK = φ ◇────────┤
      │ Decrement│        │                     ◇─────────◇         │
      │   BC     │        │                         │ Yes           │
      └─────────┘         │                         ▼  ◄────────────┤
           │  ◄──────┐    │                     ◇─────────◇  No      │
           ▼         │    │                    ◇ BPCLK = 1 ◇────────┘
      ◇─────────◇    │    │                     ◇─────────◇
     ◇ BPCLK = φ? ◇──┘    │                         │ Yes
      ◇─────────◇         │                         ▼
           │  ◄──────┐    │                     ┌─────────┐
           ▼         │    │                     │ Output  │
      ◇─────────◇    │    │                     │ Stop    │
     ◇ BTCLK = 1 ◇───┘    │                     └─────────┘
      ◇─────────◇         │                         │  ◄── Wait for ACK)
           │              │                         ▼
           ▼              │                     ╭─────────╮
      ┌─────────┐         │                     │ Return  │
      │ Output  │         │                     ╰─────────╯
      │ Data Bit│         │
      └─────────┘         │
           │              │
           ▼              │
    No ◇─────────◇ Yes    │
   ────◇ BC = φ ? ◇───────┘
        ◇─────────◇
```
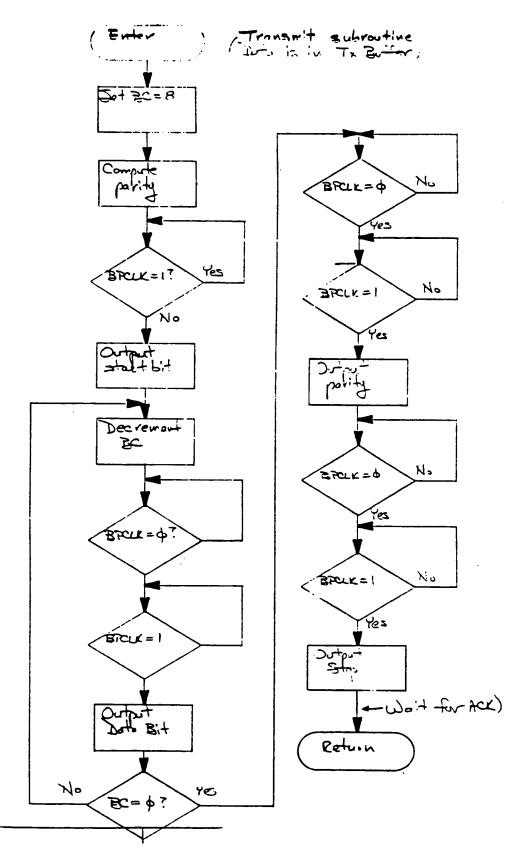
-B9-

Charge level lookup table:

Input variables: 1. Temperature
                 2. Charge voltage
                 3. Fuel guage

1. Temperature : table increments of 4°c × 16 steps = 64°c, -14°c to +50°c
   . temperatures below - 14°c use - 14°c value
   temperatures above +50°c use +50°c value

2. Charge voltage : table increments of 1.28 volts ×8 steps = 10.24Volts
   $7.\phi < V_{CHG} < 17.24$ v.

   voltages below 7 volts or above 17.24 volts will cause the charge level to be turned off and an error condition to be transmitted to the terminal processor

3. Fuel guage : 4 steps :
                 0 - 25%
                 25 - 50%
                 50 - 75%
                 75 - 100%

   TABLE OUTPUT: 4 bits, binary weighted
                 charge level = 32 ma /step
                 $0 \leq I_{CHG} \leq 480$ ma

   TABLE SIZE:

        16   ×   8   ×   2   =   256 Bytes
        ①   ②   ③
       Temp   Chg voltge %c (4 nibbles)